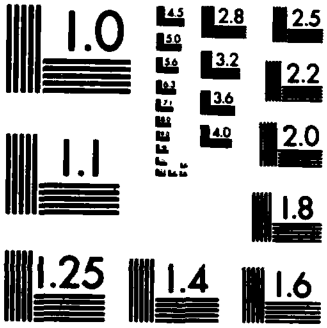
AD-A134 236 FPP1 A FLOATING POINT PACKAGE FOR PDP-8 COMPUTERS(U) 1/1
ARMY MATERIALS AND MECHANICS RESEARCH CENTER WATERTOWN
MA C B WALKER AUG 83 AMMRC-TR-83-47

UNCLASSIFIED F/G 9/2 NL

END

FILMED


1.0
4.5
5.0
5.6
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

(12)

AMMRC TR 83-47

AD

AD-A134236

FPP1, A FLOATING POINT PACKAGE FOR PDP-8 COMPUTERS

CHRISTOPHER B. WALKER
METALS RESEARCH DIVISION

August 1983

Approved for public release; distribution unlimited.

ARMY MATERIALS AND MECHANICS RESEARCH CENTER
Watertown, Massachusetts 02172

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

Mention of any trade names or manufacturers in this report shall not be construed as advertising nor as an official indorsement of approval of such products or companies by the United States Government.

DISPOSITION INSTRUCTIONS

When this report is no longer needed, Department of the Army organizations will destroy it in accordance with the procedures given in AR 380-5. Navy and Air Force elements will destroy it in accordance with applicable directions. Department of Defense contractors will destroy the report according to the requirements of Section 14 of the Industrial Security Manual for Safeguarding Classified Information. All others will return the report to U. S. Army Materials and Mechanics Research Center.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AMMRC TR 83-47	2. GOVT ACCESSION NO. A134236	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (*and Subtitle*) FPP1, A FLOATING POINT PACKAGE FOR PDP-8 COMPUTERS		5. TYPE OF REPORT & PERIOD COVERED Final Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(*s*) Christopher B. Walker		8. CONTRACT OR GRANT NUMBER(*s*)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Army Materials and Mechanics Research Center Watertown, Massachusetts 02172 DRXMR-MM		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS D/A Project:1L161102AH42
11. CONTROLLING OFFICE NAME AND ADDRESS U. S. Army Materiel Development and Readiness Command, Alexandria, Virginia 22333		12. REPORT DATE August 1983
		13. NUMBER OF PAGES 32
14. MONITORING AGENCY NAME & ADDRESS(*if different from Controlling Office*)		15. SECURITY CLASS. (*of this report*) Unclassified
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (*of this Report*)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (*of the abstract entered in Block 20, if different from Report*)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (*Continue on reverse side if necessary and identify by block number*)

Computer program documentation
Computer programs
Computers
Minicomputers
Mathematical programming
DEC PDP-8 minicomputer

20. ABSTRACT (*Continue on reverse side if necessary and identify by block number*)

This report describes a 23-bit floating point package for PDP-8 computers developed from the DEC YQ4B package that retains most of the capabilities of YQ4B while requiring only 1101 words storage.

CONTENTS

[illegible]

a 23- [illegible]

INTRODUCTION

~~The AMMRC~~ floating point (FP) package, FPP1, for PDP-8 computers was developed from the Digital Equipment Corporation (DEC) package, YQ4B (4/70), with the primary aim of reducing its size, so that an adequate calculation and control software package could be fit into the 4K memory of a PDP-8 that was used to control an X-ray diffractometer in a wide variety of experiments in the AMMRC program of materials research. It should be particularly useful for systems with memory size limitations, as in the older PDP-8's or as part of an arithmetic package committed to ROM in modern units. It follows essentially the same approach and basic procedures as YQ4B, the major changes being the exclusion of E-format input and output (I/O) and the use of a different square-root algorithm, and it retains the same accuracy, generally six decimal digits in normal computations. It has the same mnemonic and numerical codes for the basic FP instructions and function subroutines and many of the same page 0 locations and address tags as the DEC package, so it should be compatible with minimal changes with many programs using a 4K memory. The linkages for multiple-field operations are left to those users to supply.

FPP1 includes all the operations of YQ4B, excluding E-format I/O, plus two-word fix and float routines and an improved output roundoff. It occupies locations 0007, 0015, 0040 through 0064, and 5512 through 7577, for a total of 1101 words, 26% less than the 1492 words of YQ4B, the saving being approximately 10% of a 4K memory. It is not easily relocatable; a number of subroutines, constants, and storage registers are positioned so that their indirect addresses occur as regular program words (e.g., JMP I XXX) to reduce storage requirements. Its data I/O to the outside world is handled by two subroutines, READIN and TYPOUT, coded here for a teletype (TTY).

FPP1 is a package of 41 subroutines, ranging from simple register shifts to function series expansions involving other subroutines including the FP system itself, which allows the user to program mathematical computations in FP arithmetic through a set of simple block instructions, each of which calls up the detailed programming necessary for that complete operation. We shall first outline the general features of the package and discuss how to use it. At the end we add a few general comments and give a complete PAL III pass 3 listing of the assembled package for those interested in the programming details.

GENERAL

In the FPP1/YQ4B convention a normalized FP number has the form,

$$X = \text{Mantissa} \cdot 2^{\text{Exponent}} ,$$

where the Mantissa, a signed fraction with $0.5 \leq |\text{Mantissa}| < 1.0$, is specified in two words by a sign and 23 significant binary bits, and the Exponent, a signed binary integer with $|\text{Exponent}| < 2048$, is specified in two's-complement form in one word. It is stored as three consecutive words: the Exponent, then the high-order (more significant) and the low-order Mantissa words. The Exponent address serves as the address of that FP number.

To work with these three-word FP numbers, FPP1/YQ4B uses a three-word 'pseudo' floating accumulator (FAC), consisting of locations 0044, 0045, and 0046, labelled EXP, HORD, and LORD, respectively. To maintain accuracy, some operations include at intermediate stages a third word for still lower order Mantissa bits 24 thru 35; that third Mantissa word for FAC is location 0047, labelled OVER2.

FP operations generally involve doing something to or with FAC. (A reference to manipulation of a register or address simply means a manipulation of its contents.) When the operation involves another FP quantity, e.g., adding Y to FAC, that quantity is first brought to the operand register (OP), and then the operation is carried out between OP and FAC. The OP consists of locations 0040, 0041, and 0042, labelled EX1, AC1H, and AC1L, respectively, with a third word available for OP Mantissa bits 24 through 35 at location 0043, labelled OVER1.

To carry out FP operations at some point in a program, the first step is a jump to the FP Interpreter subroutine. Each succeeding word in the program is then interpreted as a FP instruction, and this mode continues until the FP Exit instruction is reached, which returns the computer to normal interpretation/operation at the next program word.

The FP Interpreter subroutine begins at location 7000. That address is stored at location 0007, so the usual jump entry to the Interpreter is obtained by the instruction: JMS I 0007 . Entrance to the FP mode clears the computer AC, but it does not affect FAC.

FP INSTRUCTIONS

There are seven FP instructions that carry out the basic add/subtract/multiply/divide/get/put/normalize operations. They are written in the usual PDP-8 memory reference instruction format: bits 0, 1, and 2 contain a non-zero operation code, and the remaining bits specify the relevant address in the standard manner (bit 3 for indirect address, bit 4 for current page vs page 0, and bits 5 through 11 for the page address). These basic FP instructions, referencing an address Y, are:

Mnemonic	Op Code	Operation
FADD Y	1	Y is added to FAC, and the result left in FAC.
FSUB Y	2	Y is subtracted from FAC, and the result left in FAC.
FMPY Y	3	FAC is multiplied by Y, and the result left in FAC.
FDIV Y	4	FAC is divided by Y, and the result left in FAC.
FGET Y	5	Y is loaded into FAC.
FPUT Y	6	FAC is stored in Y.
FNOR	7	FAC is adjusted to normalized FP form.

Operations 1 through 5 leave Y unchanged, and operation 6 leaves FAC unchanged. Operation 7 involves only FAC; the address part of this instruction is irrelevant.

A set of other instructions provide mathematical functions, input, output, and other manipulative operations that deal primarily with FAC and do not require a coded address. The complete octal code for these 'subroutine' instructions has the form, 00XY, where XY locates the appropriate entry in a table of subroutine addresses. These other FP instructions are:

Mnemonic		Octal Code	Operation
SQUARE	=	0001	FAC is squared, and the result left in FAC.
SQROOT	=	0002	The square-root of FAC is calculated, and the result left in FAC. A negative or zero FAC causes a program halt.
SIN	=	0003	The sine of FAC (assumed in radians) is calculated, and the result left in FAC.
COS	=	0004	The cosine of FAC (assumed in radians) is calculated, and the result left in FAC.
ARCTAN	=	0005	The arctangent of FAC is calculated, and the result (in radians, limited to the range, $\pm \pi/2$) left in FAC.
EXP1	=	0006	The exponentiation, e^{FAC}, is calculated, and the result left in FAC.
LOG	=	0007	The natural logarithm of FAC is calculated, and the result left in FAC. A negative or zero FAC causes a program halt.
INPUT	=	0010	A numerical quantity is entered on the TTY and converted to normalized FP form, and the result left in FAC. A detailed discussion of the input process is given in the next section.
OUTPUT	=	0011	FAC is output in decimal format on the TTY, with FAC lost afterwards. A detailed discussion of the output process is given in a later section.
NEGATE	=	0012	FAC is negated, and the result left in FAC.
FLOAT	=	0013	The two-word binary integer in locations 0057, 0060 is converted to normalized FP form, and the result left in FAC.
INVERT	=	0014	The reciprocal of FAC is calculated, and the result left in FAC. A zero FAC causes a program halt.
FIX	=	0015	FAC is converted and rounded off to a two-word binary integer, and the result left in HORD, LORD.

The addition of further FP instructions is discussed in a later section.

Finally, there is the FP Exit instruction:

FEXT	=	0000	Exit FP mode, return to normal word interpretation.

This exit clears the computer AC, but it does not affect FAC.

These mnemonics of course must either be defined in the program or included in the PAL Assembler symbol table if a program using them is to assemble properly.

The use of these FP instructions is illustrated by a PAL III pass 3 listing of a program to read in two sides of a right triangle and output its hypotenuse:

```
                 *0400
0400  4407                  JMS I 0007
0401  0010                  INPUT          /INPUT A
0402  0001                  SQUARE
0403  6213                  FPUT X
0404  0010                  INPUT          /INPUT B
0405  0001                  SQUARE
0406  1213                  FADD X
0407  0002                  SQROOT
0410  0011                  OUTPUT         /OUTPUT C
0411  0000                  FEXT
0412  7402                  HLT
0413  0000  X,              0              /THREE-WORD STORAGE
0414  0000                  0
0415  0000                  0
```

For other examples, see FPP1 subroutines such as SQARE (6772) and SQROT (7156).

INPUT

The FP instruction, INPUT, causes a program jump to the subroutine, INPVT (7400), which accepts and converts a numerical value entered on the TTY. That numerical value must be entered in fixed-point or integer format; E-format is not accepted. An input character string then has the general form: optional initial spaces (ignored); a possible sign (+ is not needed and is ignored); the decimal digits of the value being entered, with appropriately positioned decimal point (e.g., 123.45); and, finally, a terminator, which is any character except initial spaces, an initial sign, a decimal digit, or a decimal point. The TTY input routine, READIN, echoes each character as printed output, giving a visual advantage to non-printing terminators such as a following space, or a carriage return (which is automatically followed by an output line feed). The terminator stops the input cycle, and the accumulated value is converted to a properly signed, normalized FP number in FAC.

Several aspects of this routine should be noted:

1. Input of a terminator before input of any decimal digits yields zero in FAC. Input of a misplaced terminator yields an incorrect value in FAC; recovery may require restarting from the previous FP Interpreter entry in the program.

2. Input of a wrong number can be corrected easily if recognized before input of a terminator; typing a 'rub-out' restarts the input cycle, erasing all prior input for that quantity.

3. A decimal point is never a terminator. Input of a second decimal point in a character string simply redefines the position of the decimal point in that number. If no decimal point is included in the character string, it is assumed to follow the last decimal digit (i.e., integer format).

4. The number entered, disregarding decimal points and initial zeroes, must be less than $2^{31} \approx 2 \cdot 10^9$ (i.e., crudely, no more than nine significant digits), or a program halt occurs. This limit, which appreciably exceeds the 23 bit precision of the FP Mantissa, defines the magnitude of the largest number that can be entered directly without program interruption. There is no corresponding smallest number limit, so one method for entering a number $\geq 2^{31}$ is to enter its inverse and then invert the result.

OUTPUT

The FP instruction, OUTPUT, causes a jump to the subroutine, OUTPVT (7200), which converts FAC to a decimal representation that is output on the TTY. The output can be in either fixed-point or integer format, but not in E-format. The format specifications are stored beforehand as two numbers: NDEC (0054), the number of digits to the right of the decimal point; and NDIG (0055), the total number of characters to be output (not counting the sign or decimal point). The string of characters output then has the general form: initial spaces, as needed to fill the format; the sign (if positive, a space); and then the decimal digits and decimal point for the FAC value, out through NDEC digits beyond the decimal point. If the output is in integer format (NDEC = 0), the decimal point is not output. The precision of the FP Mantissa allows only six significant decimal digits, so zeroes are output for any further digits required by the format. The last significant digit to be output, a function usually of the format, is rounded off approximately to the nearest integer. If FAC = $0.X \cdot 10^N$ is too large to fit the given format but could fit it if NDEC were smaller (NDIG $\geq$ N > NDIG - NDEC), then NDEC is reduced until the output can fit (i.e., NDEC = NDIG - N), and output continues in this left-shifted format until NDIG and NDEC are redefined. If FAC is too large to fit any shifted format (i.e., N > NDIG), then NDIG decimal points are output instead.

Two points should be noted:

1. A carriage return/line feed sequence is automatically sent to the TTY after each output if SWIT1 (0061) is non-zero. That sequence can be obtained in normal mode programming by the instruction, JMS I CRLF , where CRLF = 0064.

2. FAC must be stored if it is needed for later steps, since FAC is obliterated by the output process.

OTHER COMMENTS

If improper operations are encountered, a program halt occurs. There are five such halts in FPP1, as follows:

Address	Error
5562	FAC is too large to be fixed as a two-word integer.
5700	The logarithm argument is zero or negative.
6304	The divisor is zero.
7125	The FP instruction, OOXY, is not a valid code.
7445	The input number exceeds the limiting size.

Easy recovery is possible only for the last of these, where actuating CONTINUE and then entering a terminator will still yield proper input conversion.

Input and output of ASCII characters to the external world are handled by subroutines READIN (7370) and TYPOUT (7172), respectively, coded here for a TTY. Other serial I/O devices could be used by recoding these two routines.

Other address-less FP instructions allowing the coded form, OOXY, such as functions and manipulations of FAC or conditional skips, can easily be added to the present set. The necessary steps are:

1. Enter the subroutine to carry out the operation in an arbitrary, convenient region of memory.

2. Enter ADDRESS-1 of this subroutine in the first empty space in the subroutine address table - here, 5776. The position of the entry in the table defines its octal code - here, 0016.

3. Decrement MNSBR (7142) and increment TABL2 (7057).

4. Add the mnemonic code for the operation to the PAL symbol table.

The added subroutine may itself make routine use of the FP mode and the basic FP instructions, but if it is also to use FP 'subroutine' instructions (OOXY codes) care must be taken to save pointers [see, e.g., the square-root subroutine, SQROT (7156)] and to avoid overlapping use of storage registers. The subroutine address table, TABLE2 (5761 ff), presently has room for two more entries, but it can be moved to allow room for many more; the FP Interpreter is capable of identifying codes up to 0177.

Finally, it is not necessary to enter the FP mode to use these subroutines; any of them can be accessed by a normal indirect jump subroutine instruction, JMS I YYY. Only one caution is needed: if the double precision binary multiplication routine, DMULT (6200), is used, its preset return address, 7153, should be restored afterwards for possible use by a FP divide instruction.

DISCUSSION

FPP1 was developed from the DEC package, YQ4B, with the primary objective of reducing its size while retaining most of its capabilities. The effort centered on eliminating duplication and wasted space and tightening up the programming, while retaining essentially the same basic procedures and algorithms and the same computational accuracy. This compaction has been surprisingly successful; the final package, including several extra routines, requires 391 fewer words of storage, a savings of almost 10% of a 4K memory.

There are a few differences about FPP1 that should be noted:

1. E-format I/O is not available. This could present a problem for some types of calculations, but it was not at all restrictive when FPP1 was used as part of the calculation and control software for a wide range of X-ray diffraction experiments, and it saved space.

2. The square-root is calculated by the sequence of operations, logarithm/divide by two/exponentiation, which involves working in the FP mode to depth two. This may be somewhat slower than the Newton iteration approach, but it saves significant space.

3. Arctan (x), for $0.5 \leq x < 2.0$, is calculated from the expansion,

$$\tan^{-1} x = \frac{\pi}{4} + \left\{ y - \frac{1}{3} y^3 + \frac{1}{5} y^5 \;.... \right\} \quad , \text{ where } y = \frac{x - 1}{x + 1} \quad ,$$

which converges more rapidly in this range than do the usual expansions for the small x and large x ranges, and which did not require many extra words.

4. The output is rounded off [subroutine RNDOUT (5512)] by adding $0.5 \cdot 10^N$ to FAC, where $X \cdot 10^N$ is the last significant digit to be typed, which works quite well except when all six significant digits are output and the number is large. This is a cosmetic extravagance, using 33 words, which is easily deleted.

A complete PAL III pass 3 listing and symbols follow. It includes many comments to help in understanding the general FP approach and procedures and in following the detailed programming. Its listed date marks the recent revision in preparation for this documentation that added the cosmetic output roundoff and modified slightly the I/O routines; the basic package was developed several years ago and has been heavily used. The user is urged to add other changes, as time permits, as an educational exercise.

It is hoped that both the FPP1 package and its documentation will be useful to the community of PDP-8 users.

```
                    /FLOATING POINT PACKAGE - AMMRC FPP1
                    /
                    /          24 FEBRUARY 1983
                    /
                    /THIS PACKAGE WAS DEVELOPED FROM THE DEC Y048
                    /VERSION (APRIL, 1970) AND REQUIRES 26% LESS
                    /STORAGE.  THE MAJOR ADDED RESTRICTION IS THAT
                    /E-FORMAT INPUT AND OUTPUT ARE NOT PERMITTED.
                    /MODIFIED ALGORITHMS ARE USED FOR SOME FUNCTIONS.
                    /BASIC INSTRUCTION MNEMONICS AND CODES ARE
                    /UNCHANGED.  SUBROUTINES INCLUDED ARE:
                    /  SQUARE=0001
                    /  SQROOT=0002
                    /  SIN=0003
                    /  COS=0004
                    /  ARCTAN=0005
                    /  EXPT=0006
                    /  LOG=0007
                    /  INPUT=0010
                    /  OUTPUT=0011
                    /  NEGATE=0012
                    /  FLOAT=0013            [FLOAT INTEGER IN (57,60)]
                    /  INVERT=0014
                    /  FIX=0015
                    /
                    /
                    *0007
0007    7000                    7000
                    *0015
0015    0000        AUT1,       0
                    *0040
0040    0000        EX1,        0                 /OP
0041    0000        AC1H,       0
0042    0000        AC1L,       0
0043    0000        OVER1,      0
0044    0000        EXP,        0                 /FAC
0045    0000        HORD,       0
0046    0000        LORD,       0
0047    0000        OVER2,      0
0050    0000        QUOL,       0
0051    0000        LOC1,       0
0052    0000        LOC2,       0
0053    0000        LOC3,       0
0054    0010        NDEC,       0010
0055    0020        NDIG,       0020
0056    0027        NBRX,       0027
0057    0000        NBRHI,      0
0060    0000        NBRLO,      0
0061    7777        SWIT1,      7777
0062    7370        READ,       READIN
0063    7172        TYPE,       TYPOUT
0064    5724        CRLF,       CRLFRT
```

```
                *5512
5512  ØØØØ  RNDOUT,   Ø                 /ROUNDOFF OUTPUT
5513  1462            TAD I READ        /ADDS Ø.5*(1Ø**-N) TO FAC
5514  1Ø54            TAD NDEC
5515  751Ø  DVTEN2,   SPA
5516  5346            JMP OUTØ
5517  135Ø            TAD M6
552Ø  75ØØ            SMA
5521  72ØØ            CLA               /ROUNDOFF SIXTH DIGIT
5522  7Ø4Ø            CMA               /ROUNDOFF TO FORMAT
5523  135Ø            TAD M6
5524  3463            DCA I TYPE
5525  44Ø7            JMS I ØØØ7
5526  6752            FPUT I XXX
5527  5751            FGET I UNE
553Ø  ØØØØ            FEXT
5531  3Ø44            DCA EXP
5532  741Ø            SKP
5533  4715            JMS I DVTEN2
5534  2463            ISZ I TYPE
5535  5333            JMP .-2
5536  44Ø7            JMS I ØØØ7
5537  1752            FADD I XXX
554Ø  ØØØØ            FEXT
5541  1Ø44            TAD EXP
5542  775Ø            SPA SNA CLA
5543  5712            JMP I RNDOUT
5544  4715            JMS I DVTEN2
5545  2462            ISZ I READ
5546  72ØØ  OUTØ,     CLA
5547  5712            JMP I RNDOUT
555Ø  7772  M6,       -6
5551  5666  UNE,      5666
5552  5734  XXX,      5734
5553  ØØØØ  FIXX,     Ø                 /FIX FAC
5554  1Ø44            TAD EXP           /EXIT WITH INTEGER IN (HORD,LORD)
5555  751Ø            SPA
5556  724Ø            CLA CMA           /[FAC] < Ø.5: SHIFT 3Ø(8)
5557  7Ø41            CMA IAC
556Ø  1Ø56            TAD NBRX          /[FAC] > Ø.5: SHIFT [27(8)-EXP]
5561  751Ø            SPA
5562  74Ø2            HLT               /[FAC] > 2**23: TOO LARGE
5563  716Ø            CLL CML CMA
5564  3776            DCA I AMNT5
5565  2776            ISZ I AMNT5       /WAS EXP = 27(8)?
5566  4777            JMS I RSHFT3      /NO: RIGHT SHIFT MANTISSA
5567  1Ø47            TAD OVER2         /ROUND OFF TO NEAREST INTEGER
557Ø  771Ø            SPA CLA
5571  2Ø46            ISZ LORD
5572  5753            JMP I FIXX
5573  2Ø45            ISZ HORD
5574  7ØØØ            NOP
5575  5753            JMP I FIXX
5576  6655  AMNT5,    AMOUNT
5577  667Ø  RSHFT3,   RSHFT
```

```
5600  0000  ACNEG,    0                    /NEGATE FAC
5601  7300            CLA CLL
5602  1047            TAD OVER2
5603  7041            CMA IAC
5604  3047            DCA OVER2
5605  1046            TAD LORD
5606  7040            CMA
5607  7430            SZL
5610  7101            CLL IAC
5611  3046            DCA LORD
5612  1045            TAD HORD
5613  7040            CMA
5614  7430            SZL
5615  7101            CLL IAC
5616  3045            DCA HORD
5617  5600            JMP I ACNEG
5620  0000  FEXP,     0                    /F.P. EXPONENTIAL
5621  1045            TAD HORD
5622  7510            SPA
5623  4200            JMS ACNEG            /MAKE FAC POSITIVE
5624  3275            DCA FLOG             /NON-ZERO IF FAC WAS > 0
5625  4252            JMS SETODD
5626  4641            JMS I SETEV
5627  3252            DCA SETODD
5630  3664            DCA I SERSV
5631  7350            CLL CLA CMA RAR
5632  4674            JMS I SERIEZ
5633  1225            TAD FEXP+5
5634  3664            DCA I SERSV
5635  1275            TAD FLOG
5636  7650            SNA CLA              /WAS FAC NEGATIVE?
5637  4665            JMS I INVRT1         /YES, INVERT RESULT
5640  5620            JMP I FEXP           /NO
5641  6042  SETEV,    SETEVN
5642  0000  XMP1,     0                    /MAKES (X-1)/(X+1)
5643  4407            JMS I 0007
5644  1266            FADD ONE
5645  6334            FPUT X
5646  2673            FSUB I DEUX
5647  4334            FDIV X
5650  0000            FEXT
5651  5642            JMP I XMP1
5652  0000  SETODD,   0                    /SETS TERMS FOR SERIES
5653  4407            JMS I 0007           /BEGINNING WITH X
5654  6334            FPUT X
5655  6337            FPUT TR1
5656  6345            FPUT TR3
5657  5266            FGET ONE
5660  6342            FPUT TR2
5661  6350            FPUT TR4
5662  0000            FEXT
5663  5652            JMP I SETODD
5664  6150  SERSV,    SERIES+3
5665  6137  INVRT1,   INVRT
```

```
5666  0001  ONE,      0001
5667  2000            2000
5670  0000  LOGG2,    0000                 /LOG 2
5671  2613            2613
5672  4414            4414
5673  6134  DEUX,     TWO
5674  6145  SERIEZ,   SERIES
5675  0000  FLOG,     0                    /F.P. LOGARITHM
5676  1045            TAD WORD
5677  7750            SPA SNA CLA
5700  7402            HLT                  /NEGATIVE ARGUMENT
5701  1044            TAD EXP
5702  3060            DCA NBRLO
5703  3044            DCA EXP              /0.5 < X < 1.0
5704  4242            JMS XMP1
5705  4252            JMS SETODD
5706  3252            DCA SETODD
5707  1007            TAD 0007
5710  4674            JMS I SERIEZ
5711  1060            TAD NBRLO
5712  7710            SPA CLA
5713  7040            CMA
5714  3057            DCA NBRHI
5715  4353            JMS FLOWT
5716  4407            JMS I 0007
5717  3270            FMPY LOGG2
5720  1345            FADD TR3
5721  1345            FADD TR3
5722  0000            FEXT
5723  5675            JMP I FLOG
5724  0000  CRLFRT,   0
5725  1332            TAD C215
5726  4463            JMS I TYPE
5727  1333            TAD C212
5730  4463            JMS I TYPE
5731  5724            JMP I CRLFRT
5732  0215  C215,     0215
5733  0212  C212,     0212                 /DO NOT RELOCATE
5734  0000  X,        0
5735  0000            0
5736  0000            0
5737  0000  TR1,      0                    /OUTPUT BUFFER
5740  0000            0
5741  0000            0
5742  0000  TR2,      0
5743  0000            0
5744  0000            0
5745  0000  TR3,      0
5746  0000            0
5747  0000            0
5750  0000  TR4,      0
5751  0000            0
5752  0000            0
5753  0000  FLOWT,    0                    /TWO WORD INTEGER FLOAT
5754  4407            JMS I 0007
5755  5056            FGET NBRX
```

```
5756  7000              FNOR
5757  0000              FEXT
5760  5753              JMP I FLOWT
5761  6771  TABLE2,     SQARE-1           /SUBROUTINE ADDRESS TABLE
5762  7155              SQROT-1
5763  5777              FSIN-1
5764  6033              FCOS-1
5765  6065              ARCTN-1
5766  5617              FEXP-1
5767  5674              FLOG-1
5770  7377              INPUT-1
5771  7177              OUTPUT-1
5772  5577              ACNEG-1
5773  5752              FLOWT-1
5774  6136              INVRT-1
5775  5552              FIXX-1
5776  0000              0
5777  0000              0

            *6000
6000  0000  FSIN,       0                 /F.P. SINE
6001  1045              TAD HORD
6002  7700              SMA CLA
6003  5206              JMP .+3
6004  4633              JMS I ACNEG1
6005  7126              CLL CML RTL
6006  3266              DCA ARCTN
6007  1044              TAD EXP
6010  7750              SPA SNA CLA       /IS MAG. FAC < 1?
6011  5217              JMP .+6           /YES
6012  4407              JMS I 0007        /NO, REDUCE BY PI/2.
6013  2372              FSUB HALFPI
6014  0000              FEXT
6015  2266              ISZ ARCTN
6016  5207              JMP .-7
6017  4665              JMS I SETOD       /SET FOR SINE SERIES
6020  1266              TAD ARCTN
6021  7010              RAR
6022  3266              DCA ARCTN
6023  7430              SZL               /ODD MULTIPLE OF PI/2 SUBTRACTED?
6024  4242              JMS SETEVN        /YES, SET FOR COSINE SERIES
6025  7350              CLL CLA CMA RAR
6026  4345              JMS SERIES
6027  1266              TAD ARCTN
6030  7010              RAR
6031  7630              SZL CLA           /SIGN REVERSAL REQUIRED?
6032  4633              JMS I ACNEG1      /YES
6033  5600  ACNEG1,     JMP I FSIN
6034  0000  FCOS,       0                 /F.P. COSINE
6035  4407              JMS I 0007
6036  1372              FADD HALFPI
6037  0000              FEXT
6040  4200              JMS FSIN
6041  5634              JMP I FCOS
6042  0000  SETEVN,     0                 /SETS TERMS FOR SERIES
6043  4407              JMS I 0007        /BEGINNING WITH 1
```

```
6Ø44   6744            FPUT I TS1        /USE AFTER JMS SETODD
6Ø45   6771            FPUT I TS3
6Ø46   2771            FSUB I TS3
6Ø47   6776            FPUT I TS4
6Ø5Ø   ØØØØ            FEXT
6Ø51   5642   XXMP1,   JMP I SETEVN
6Ø52   ØØØØ   TERM,    Ø                 /INCREMENTS TS4, MAKES TS4! IN TS2
6Ø53   44Ø7            JMS I ØØØ7        /AND MAKES X**TS4 IN TS1
6Ø54   5776            FGET I TS4
6Ø55   1733            FADD I UNO
6Ø56   6776            FPUT I TS4
6Ø57   3777   TRMSW,   FMPY I TS2
6Ø6Ø   6777            FPUT I TS2
6Ø61   5744            FGET I TS1
6Ø62   3775            FMPY I XX
6Ø63   6744            FPUT I TS1
6Ø64   ØØØØ            FEXT
6Ø65   5652   SETOD,   JMP I TERM
6Ø66   ØØØØ   ARCTN,   Ø                 /F.P. ARCTANGENT
6Ø67   1Ø45            TAD HORD
6Ø7Ø   745Ø            SNA
6Ø71   5666            JMP I ARCTN
6Ø72   751Ø            SPA
6Ø73   4633            JMS I ACNEG1
6Ø74   3234            DCA FCOS
6Ø75   1Ø44            TAD EXP
6Ø76   75ØØ            SMA
6Ø77   53Ø2            JMP .+3
61ØØ   4665            JMS I SETOD       /CASE I: FAC < Ø.5
61Ø1   5326            JMP READY
61Ø2   711Ø            CLL RAR
61Ø3   764Ø            SZA CLA
61Ø4   5316            JMP BIGG
61Ø5   4651            JMS I XXMP1       /CASE II: Ø.5 < FAC < 2.Ø
61Ø6   4665            JMS I SETOD
61Ø7   44Ø7            JMS I ØØØ7
611Ø   5372            FGET HALFPI
6111   4334            FDIV TWO
6112   1771            FADD I TS3
6113   6771            FPUT I TS3
6114   ØØØØ            FEXT
6115   5326            JMP READY
6116   4337   BIGG,    JMS INVRT         /CASE III: 2.Ø < FAC
6117   4633            JMS I ACNEG1
612Ø   4665            JMS I SETOD
6121   44Ø7            JMS I ØØØ7
6122   5372            FGET HALFPI
6123   1771            FADD I TS3
6124   6771            FPUT I TS3
6125   ØØØØ            FEXT
6126   1ØØ7   READY,   TAD ØØØ7
6127   4345            JMS SERIES
613Ø   1234            TAD FCOS
6131   765Ø            SNA CLA
6132   4633            JMS I ACNEG1
6133   5666   UNO,     JMP I ARCTN
```

```
6134  0002  TWO,      0002
6135  2000            2000
6136  0000            0000
6137  0000  INVRT,    0                    /MAKES RECIPROCAL OF FAC
6140  4665            JMS I SETOD
6141  4407            JMS I 0007
6142  4775            FDIV I XX
6143  0000            FEXT
6144  5737  TS1,      JMP I INVRT
6145  0000  SERIES,   0                    /FORMS POWER SERIES
6146  3257            DCA TRMSW            /ENTER WITH SWITCH SETTING IN AC
6147  3242            DCA SETEVN           /(3777: TS4! IN TS2: 7000: TS4)
6150  4252            JMS TERM             /OVERLAY HERE WITH 0 IF SERIES HAS
6151  4252            JMS TERM             /ALL TERMS INSTEAD OF ALTERNATE
6152  1665            TAD I SETOD          /TERMS
6153  7650            SNA CLA              /NON-ZERO SETODD: SIGN ALTERNATES
6154  5362            JMP .+6
6155  2242            ISZ SETEVN
6156  1242            TAD SETEVN
6157  7010            RAR
6160  7630            SZL CLA
6161  4633            JMS I ACNEG1
6162  4407            JMS I 0007
6163  4777            FDIV I TS2
6164  1771            FADD I TS3
6165  6771            FPUT I TS3
6166  0000            FEXT
6167  2050            ISZ QUOL
6170  5350            JMP SERIES+3
6171  5745  TS3,      JMP I SERIES
6172  0001  HALFPI,   0001                 /0.5 * PI
6173  3110            3110
6174  3755            3755
6175  5734  XX,       X
6176  5750  TS4,      TR4
6177  5742  TS2,      TR2

            *6200
6200  7153  DMULT,    MUL+2                /MULTIPLY FAC BY OP
6201  4775            JMS I AMNT1          /IS FAC = 0?
6202  5370            JMP EXIT1            /YES
6203  1265            TAD MULSW
6204  3272            DCA SNSW
6205  4261            JMS SIGN
6206  1042            TAD AC1L
6207  3051            DCA LOC1
6210  1046            TAD LORD
6211  4657            JMS I MULT           /AC1L * LORD
6212  7200            CLA
6213  1052            TAD LOC2
6214  3047            DCA OVER2
6215  1045            TAD HORD
6216  4657            JMS I MULT           /AC1L * HORD
6217  1047            TAD OVER2
6220  3047            DCA OVER2
```

```
6221  7ØØ4              RAL
6222  1Ø52              TAD LOC2
6223  3Ø53              DCA LOC3
6224  7ØØ4              RAL
6225  3Ø15              DCA AUT1
6226  1Ø41              TAD AC1H
6227  3Ø51              DCA LOC1
623Ø  1Ø46              TAD LORD
6231  4657              JMS I MULT      /AC1H * LORD
6232  1Ø47              TAD OVER2
6233  3Ø47              DCA OVER2
6234  7ØØ4              RAL
6235  1Ø52              TAD LOC2
6236  1Ø53              TAD LOC3
6237  3Ø53              DCA LOC3
624Ø  7ØØ4              RAL
6241  1Ø15              TAD AUT1
6242  3Ø15              DCA AUT1
6243  1Ø45              TAD HORD
6244  4657              JMS I MULT      /AC1H * HORD
6245  1Ø53              TAD LOC3
6246  3Ø46              DCA LORD
6247  7ØØ4              RAL
625Ø  1Ø52              TAD LOC2
6251  1Ø15              TAD AUT1
6252  3Ø45              DCA HORD
6253  466Ø              JMS I DNORM1
6254  23ØØ              ISZ SNREG
6255  56ØØ  ACNEG2,     JMP I DMULT
6256  5372              JMP FACNEG
6257  6743  MULT,       DMPY
626Ø  64ØØ  DNORM1,     DNORM
6261  ØØØØ  SIGN,       Ø               /SET SIGNS OF FAC AND OP
6262  7144              CLL CMA RAL
6263  33ØØ              DCA SNREG
6264  1Ø45              TAD HORD
6265  77ØØ  MULSW,      SMA CLA
6266  5271              JMP .+3
6267  4655              JMS I ACNEG2    /MAKE FAC POSITIVE
627Ø  23ØØ              ISZ SNREG
6271  1Ø41              TAD AC1H
6272  ØØØØ  SNSW,       Ø               /MAKE OP POSITIVE FOR MULTIPLY
6273  5661              JMP I SIGN      /MAKE OP NEGATIVE FOR DIVIDE
6274  47Ø1              JMS I NEGOP
6275  23ØØ              ISZ SNREG
6276  5661              JMP I SIGN
6277  5661              JMP I SIGN
63ØØ  ØØØØ  SNREG,      Ø
63Ø1  7Ø6Ø  NEGOP,      NEG
63Ø2  1Ø51  DIV,        TAD LOC1        /OPCODE 4
63Ø3  765Ø              SNA CLA
63Ø4  74Ø2              HLT             /OP IS ZERO
63Ø5  1Ø4Ø              TAD EX1
63Ø6  7Ø41              CMA IAC
63Ø7  3Ø4Ø              DCA EX1
631Ø  4774              JMS I UNFLW1    /UNDERFLOW?
```

```
6311  4775            JMS I AMNT1       /IS FAC = 0?
6312  5370            JMP EXIT1         /YES, UNDERFLOW OR ZERO FAC
6313  1376            TAD DIVSW
6314  3272            DCA SNSW
6315  4261            JMS SIGN
6316  7300            CLA CLL
6317  3052            DCA LOC2
6320  3050            DCA QUOL
6321  1377            TAD CN23
6322  3053            DCA LOC3
6323  1042  DVX,      TAD AC1L          /SUBTRACT DIVISOR FROM DIVIDEND
6324  1046            TAD LORD
6325  3051            DCA LOC1
6326  7004            RAL
6327  1041            TAD AC1H
6330  1045            TAD HORD
6331  7420            SNL               /WAS DIVIDEND LARGER THAN DIVISOR?
6332  5336            JMP .+4
6333  3045            DCA HORD          /YES: DIFFERENCE IS NEW DIVIDEND
6334  1051            TAD LOC1
6335  3046            DCA LORD
6336  7200            CLA
6337  1050            TAD QUOL          /ROTATE (LOC2,QUOL) LEFT ONE
6340  7004            RAL
6341  3050            DCA QUOL
6342  1052            TAD LOC2
6343  7004            RAL
6344  3052            DCA LOC2
6345  1046            TAD LORD          /SHIFT DIVIDEND LEFT ONE
6346  7004            RAL
6347  3046            DCA LORD
6350  1045            TAD HORD
6351  7004            RAL
6352  3045            DCA HORD
6353  2053            ISZ LOC3
6354  5323            JMP DVX
6355  1042            TAD AC1L          /SET UP ROUNDOFF
6356  1046            TAD LORD
6357  7204            CLA RAL
6360  1041            TAD AC1H
6361  1045            TAD HORD
6362  7210            CLA RAR
6363  3047            DCA OVER2
6364  1050            TAD QUOL
6365  3046            DCA LORD
6366  1052            TAD LOC2
6367  3045            DCA HORD
6370  4660  EXIT1,    JMS I DNORM1
6371  2300            ISZ SNREG
6372  4655  FACNEG,   JMS I ACNEG2
6373  5600            JMP I DMULT
6374  6511  UNFLW1,   UNFLW
6375  6655  AMNT1,    AMOUNT
6376  7710  DIVSW,    SPA CLA
6377  7751  CN23,     7751
```

```
                    *6400
6400   0000   DVORM,      0                  /NORMALIZE FAC
6401   4653               JMS I AMNT2        /IS FAC = 0?
6402   5245               JMP EXIT0          /YES
6403   1046               TAD LORD           /IS FAC APPARENT -0?
6404   7640               SZA CLA
6405   5216               JMP OK
6406   1045               TAD HORD
6407   7510               SPA
6410   7041               CMA IAC
6411   7700               SMA CLA
6412   5216               JMP OK
6413   7360               CLA CLL CMA CML
6414   3653               DCA I AMNT2
6415   4654               JMS I RSHFT1       /YES: SHIFT FAC RIGHT ONE
6416   3040   OK,         DCA EX1
6417   3041               DCA AC1H
6420   1045               TAD HORD
6421   7700               SMA CLA
6422   5225               JMP .+3
6423   2041               ISZ AC1H
6424   4652               JMS I ACNEG3       /MAKE FAC POSITIVE
6425   1045               TAD HORD
6426   7004               RAL
6427   7710               SPA CLA            /IS BIT ONE NON-ZERO?
6430   5234               JMP .+4            /YES: DONE
6431   4255               JMS LSHFT          /NO: SHIFT FAC LEFT ONE
6432   2040               ISZ EX1
6433   5225               JMP .-6
6434   1047               TAD OVER2
6435   7104               CLL RAL
6436   7204               CLA RAL
6437   4273               JMS RNDOFF         /ROUNDOFF
6440   1040               TAD EX1
6441   7040               CMA
6442   3040               DCA EX1
6443   4311               JMS UNFLW
6444   5247               JMP .+3            /NO UNDERFLOW
6445   3044   EXIT0,      DCA EXP
6446   5600               JMP I DVORM
6447   1041               TAD AC1H
6450   7640               SZA CLA
6451   4652               JMS I ACNEG3
6452   5600   ACNEG3,     JMP I DVORM
6453   6655   AMNT2,      AMOUNT
6454   6670   RSHFT1,     RSHFT
6455   0000   LSHFT,      0                  /FOUR WORD LEFT SHIFT
6456   1047               TAD OVER2
6457   7104               CLL RAL
6460   3047               DCA OVER2
6461   1046               TAD LORD
6462   7004               RAL
6463   3046               DCA LORD
6464   1045               TAD HORD
6465   7004               RAL
```

```
6466  3045            DCA HORD
6467  1273            TAD RNDOFF
6470  7004            RAL
6471  3273            DCA RNDOFF
6472  5655            JMP I LSHFT
6473  0000  RNDOFF,   0                   /ADD C(AC) TO POSITIVE FAC
6474  7100            CLL
6475  1046            TAD LORD
6476  3046            DCA LORD
6477  7004            RAL
6500  1045            TAD HORD            /IF FAC BIT 0 IS SET, SHIFT
6501  7500            SMA                 /HORD WORD RIGHT ONE
6502  5306            JMP .+4
6503  2044            ISZ EXP
6504  7000            NOP
6505  7010            RAR
6506  3045            DCA HORD
6507  3047            DCA OVER2
6510  5673            JMP I RNDOFF
6511  0000  UNFLW,    0                   /EXPONENT UNDERFLOW CHECK
6512  1044            TAD EXP
6513  7710            SPA CLA
6514  1040            TAD,EX1
6515  7710            SPA CLA
6516  7040            CMA
6517  3334            DCA ADDOP
6520  1040            TAD EX1
6521  1044            TAD EXP
6522  7001            IAC
6523  7500            SMA                 /IS SUM POSITIVE?
6524  2334            ISZ ADDOP           /WERE EXP AND EX1 BOTH NEGATIVE?
6525  5332            JMP .+5             /NO
6526  7300            CLA CLL             /YES, UNDERFLOW; SET FAC = 0
6527  3045            DCA HORD
6530  3046            DCA LORD
6531  2311            ISZ UNFLW           /INCREMENT POINTER
6532  3044            DCA EXP
6533  5711            JMP I UNFLW
6534  0000  ADDOP,    0                   /ADD OP TO (RNDOFF,FAC)
6535  7300            CLA CLL
6536  1047            TAD OVER2
6537  1043            TAD OVER1
6540  3047            DCA OVER2
6541  7004            RAL
6542  1046            TAD LORD
6543  1042            TAD AC1L
6544  3046            DCA LORD
6545  7004            RAL
6546  1045            TAD HORD
6547  1041            TAD AC1H
6550  3045            DCA HORD
6551  7004            RAL
6552  1273            TAD RNDOFF
6553  3273            DCA RNDOFF
6554  5734            JMP I ADDOP
```

```
6555  0000  X10,      0                 /ADD C(ACNEG) TO 10 * FAC
6556  1047            TAD OVER2         /OVERFLOW IS IN RNDOFF
6557  3043            DCA OVER1
6560  1046            TAD LORD
6561  3042            DCA AC1L
6562  1045            TAD HORD
6563  3041            DCA AC1H
6564  3273            DCA RNDOFF
6565  4255            JMS LSHFT
6566  4255            JMS LSHFT
6567  4334            JMS ADDOP
6570  4255            JMS LSHFT
6571  1652            TAD I ACNEG3
6572  3043            DCA OVER1
6573  3042            DCA AC1L
6574  3041            DCA AC1H
6575  4334            JMS ADDOP
6576  1273            TAD RNDOFF
6577  5755            JMP I X10         /EXIT WITH C(RNDOFF) IN AC
            *6600
6600  0000  DADD,     0                 /ADD OP TO FAC
6601  3050            DCA QUOL
6602  4255            JMS AMOUNT        /IS FAC = 0?
6603  5230            JMP GETOP         /YES: GET OP
6604  1051            TAD LOC1
6605  7650            SNA CLA           /IS OP = 0?
6606  5600            JMP I DADD        /YES: DONE
6607  1040            TAD EX1           /ARE EXPONENTS EQUAL?
6610  7041            CMA IAC
6611  1044            TAD EXP
6612  7450            SNA
6613  5245            JMP OFFSET        /YES
6614  3270            DCA RSHFT         /NO: CAN OP AND FAC BE ALIGNED?
6615  1270            TAD RSHFT
6616  7500            SMA
6617  7041            CMA IAC
6620  3255            DCA AMOUNT
6621  1255            TAD AMOUNT
6622  1327            TAD C24
6623  7700            SMA CLA
6624  5241            JMP ALIGN         /YES
6625  1270            TAD RSHFT         /NO: WHICH IS LARGER?
6626  7700            SMA CLA
6627  5236            JMP GETOP+6       /FAC
6630  1040  GETOP,    TAD EX1           /OP: GET OP
6631  3044            DCA EXP
6632  1041            TAD AC1H
6633  3045            DCA HORD
6634  1042            TAD AC1L
6635  3046            DCA LORD
6636  7040            CMA
6637  3050            DCA QUOL          /IF FAC = 0 OR IF NO ALIGNMENT
6640  5600            JMP I DADD        /POSSIBLE, SET QUOL = 7777
6641  1270  ALIGN,    TAD RSHFT
6642  7004            RAL
```

```
6643  7200            CLA
6644  4270            JMS RSHFT
6645  7340  OFFSET,   CLA CLL CMA        /SHIFT FAC AND OP RIGHT ONE
6646  3255            DCA AMOUNT
6647  4270            JMS RSHFT
6650  7360            CLA CLL CMA CML
6651  3255            DCA AMOUNT
6652  4270            JMS RSHFT
6653  4732            JMS I ADDUP
6654  5600            JMP I DADD
6655  0000  AMOUNT,   0                  /TEST IF FAC = 0
6656  1045            TAD HORD
6657  7640            SZA CLA            /IF YES, EXIT NORMALLY
6660  5266            JMP .+6
6661  1046            TAD LORD
6662  7640            SZA CLA
6663  5266            JMP .+3
6664  1047            TAD OVER2
6665  7640            SZA CLA
6666  2255            ISZ AMOUNT         /IF NO, EXIT TO SECOND INSTRUCTION
6667  5655            JMP I AMOUNT
6670  0000  RSHFT,    0                  /THREE WORD RIGHT SHIFT
6671  7420            SNL                /IF LINK = 1, SHIFT FAC
6672  1331            TAD TAG2           /IF LINK = 0, SHIFT OP
6673  1330            TAD TAG1
6674  3051            DCA LOC1
6675  1255            TAD AMOUNT         /NEGATIVE OF AMOUNT OF SHIFT
6676  7041            CMA IAC
6677  1451            TAD I LOC1
6700  3451            DCA I LOC1         /EXPONENT SHIFTED
6701  2051            ISZ LOC1           /SET UP MANTISSA ADDRESSES
6702  1051            TAD LOC1
6703  7001            IAC
6704  3052            DCA LOC2
6705  1052            TAD LOC2
6706  7001            IAC
6707  3053            DCA LOC3
6710  7100  SHIFT,    CLL
6711  1451            TAD I LOC1
6712  7510            SPA
6713  7020            CML
6714  7010            RAR
6715  3451            DCA I LOC1
6716  1452            TAD I LOC2
6717  7010            RAR
6720  3452            DCA I LOC2
6721  1453            TAD I LOC3
6722  7010            RAR
6723  3453            DCA I LOC3
6724  2255            ISZ AMOUNT
6725  5310            JMP SHIFT
6726  5670            JMP I RSHFT
6727  0030  C24,      0030
6730  0044  TAG1,     EXP
6731  7774  TAG2,     EX1-EXP
6732  6534  ADDUP,    ADDOP
```

```
6733  7117  TABLE1,   SBR-1          /OPCODE ADDRESS TABLE
6734  7143            ADD-1
6735  7142            SUB-1
6736  7150            MUL-1
6737  6301            DIV-1
6740  7077            GET-1
6741  7106            PUT-1
6742  7144            NRM-1
6743  0000  DMPY,     0              /MULTIPLY AC BY LOC1
6744  3050            DCA QUOL
6745  3052            DCA LOC2
6746  1371            TAD CN12
6747  3255            DCA AMOUNT
6750  7100            CLL
6751  1050            TAD QUOL
6752  7010            RAR
6753  3050            DCA QUOL
6754  1052            TAD LOC2
6755  7420            SNL
6756  5361            JMP .+3
6757  7100            CLL
6760  1051            TAD LOC1
6761  7010            RAR
6762  3052            DCA LOC2
6763  2255            ISZ AMOUNT
6764  5351            JMP DMPY+6
6765  1050            TAD QUOL       /EXIT WITH LESS SIGNIFICANT
6766  7010            RAR            /TWELVE BITS IN AC AND MORE
6767  7100            CLL            /SIGNIFICANT BITS IN LOC2
6770  5743            JMP I DMPY
6771  7764  CN12,     7764
6772  0000  SQARE,    0              /F.P. SQUARE
6773  4407            JMS I 0007
6774  6770            FPUT I CN12-1
6775  3770            FMPY I CN12-1
6776  0000            FEXT
6777  5772            JMP I SQARE

            *7000
7000  0000  FPNT,     0              /F.P. INTERPRETER
7001  7600            7600
7002  3043            DCA OVER1
7003  3047            DCA OVER2
7004  3051            DCA LOC1
7005  1600            TAD I FPNT     /GET ADDRESS REFERENCED
7006  0212            AND .+4
7007  7650            SNA CLA        /PAGE 0?
7010  5213            JMP .+3        /YES
7011  1201            TAD FPNT+1
7012  0200            AND FPNT
7013  3260            DCA NEG        /BITS 0,--4 GIVE PAGE ID
7014  1201            TAD FPNT+1
7015  7040            CMA
7016  0600            AND I FPNT
7017  1260            TAD NEG
7020  3260            DCA NEG        /ADDRESS
```

```
7021  1255            TAD M1
7022  0600            AND I FPNT
7023  7650            SNA CLA         /INDIRECT?
7024  5227            JMP .+3         /NO
7025  1660            TAD I NEG       /YES: GET DIRECT ADDRESS
7026  3260            DCA NEG
7027  1260            TAD NEG         /MOVE F.P. WORD FROM ADDRESS TO OP
7030  3015            DCA AUT1        /AND TEST IF OP = 0
7031  1660            TAD I NEG
7032  3040            DCA EX1
7033  1415            TAD I AUT1
7034  7440            SZA
7035  2051            ISZ LOC1        /IF OP = 0, LOC1 = 0
7036  3041            DCA AC1H
7037  1415            TAD I AUT1
7040  7440            SZA
7041  2051            ISZ LOC1
7042  3042            DCA AC1L
7043  1600            TAD I FPNT      /GET OPCODE
7044  0007            AND 0007
7045  7106            CLL RTL
7046  7006            RTL
7047  1256            TAD TABL1       /GET ENTRY FROM TABLE
7050  3015            DCA AUT1
7051  1415            TAD I AUT1
7052  3015            DCA AUT1
7053  2200            ISZ FPNT
7054  5415            JMP I AUT1      /GO THERE
7055  0400  M1,       0400
7056  6732  TABL1,    TABLE1-1
7057  5774  TABL2,    TABLE2+13       /TABLE2 - MNSBR - 2
7060  0000  NEG,      0               /NEGATE OP
7061  7300            CLA CLL
7062  1043            TAD OVER1
7063  7041            CMA IAC
7064  3043            DCA OVER1
7065  1042            TAD AC1L
7066  7040            CMA
7067  7430            SZL
7070  7101            CLL IAC
7071  3042            DCA AC1L
7072  1041            TAD AC1H
7073  7040            CMA
7074  7430            SZL
7075  7101            CLL IAC
7076  3041            DCA AC1H
7077  5660            JMP I NEG
7100  1040  GET,      TAD EX1         /OPCODE 5
7101  3044            DCA EXP
7102  1041            TAD AC1H
7103  3045            DCA HORD
7104  1042            TAD AC1L
7105  3046            DCA LORD
7106  5201            JMP FPNT+1
7107  1044  PUT,      TAD EXP         /OPCODE 6
7110  3660            DCA I NEG
```

```
7111  1260              TAD NEG
7112  3015              DCA AUT1
7113  1045              TAD HORD
7114  3415              DCA I AUT1
7115  1046              TAD LORD
7116  3415              DCA I AUT1
7117  5201              JMP FPNT+1
7120  1260  SBR,        TAD NEG             /OPCODE 0
7121  7450              SNA                 /EXIT?
7122  5600              JMP I FPNT          /YES
7123  1342              TAD MNSBR           /NO: CHECK CODE
7124  7540              SMA SZA
7125  7402              HLT                 /UNDEFINED SUBROUTINE
7126  1257              TAD TABL2           /GET ENTRY FROM TABLE
7127  3015              DCA AUT1
7130  1415              TAD I AUT1
7131  3015              DCA AUT1
7132  1200              TAD FPNT            /SAVE POINTER: SUBROUTINES
7133  3341              DCA SAV1            /CAN USE F.P. TO DEPTH ONE
7134  4415              JMS I AUT1          /JMS TO SUBROUTINE
7135  7200              CLA
7136  1341              TAD SAV1            /RESET POINTER
7137  3200              DCA FPNT
7140  5201              JMP FPNT+1
7141  0000  SAV1,       0
7142  7763  MNSBR,      7763                /NEGATIVE OF NO. OF SUBROUTINES
7143  4260  SUB,        JMS NEG             /OPCODE 2
7144  4747  ADD,        JMS I FLAD          /OPCODE 1
7145  4750  NRM,        JMS I DNORM2        /OPCODE 7
7146  5201              JMP FPNT+1
7147  6600  FLAD,       DADD
7150  6400  DNORM2,     DNORM
7151  4754  MUL,        JMS I UNFLW2        /OPCODE 3
7152  4755              JMS I MPLY
7153  5201              JMP FPNT+1
7154  6511  UNFLW2,     UNFLW
7155  6200  MPLY,       DMULT
7156  0000  SQROT,      0                   /F.P. SQUARE ROOT
7157  1341              TAD SAV1            /SHIFT POINTER TO USE LOG
7160  3372              DCA TYPOUT          /AND EXP SUBROUTINES HERE
7161  4407              JMS I 0007
7162  0007              LOG
7163  4771              FDIV I DOS
7164  0006              EXPT
7165  0000              FEXT
7166  1372              TAD TYPOUT
7167  3341              DCA SAV1
7170  5756              JMP I SQROT
7171  6134  DOS,        TWO
7172  0000  TYPOUT,     0
7173  6046              TLS
7174  6041              TSF
7175  5374              JMP .-1
7176  7300              CLA CLL
7177  5772              JMP I TYPOUT
```

```
                   *72ØØ
 72ØØ   ØØØØ    OUTPUT,     Ø                    /OUTPUT FAC
 7201   3047                DCA OVERP            /FORMAT SPEC. HAS BEEN STORED IN
 72Ø2   3370                DCA READIN           /NDEC (NO. OF DIGITS TO RIGHT OF
 72Ø3   1Ø45                TAD WORD             /DEC. PT) AND NDIG (TOTAL NO. OF
 7204   77ØØ                SMA CLA              /DIGITS) PRIOR TO ENTRY
 72Ø5   521Ø                JMP .+3
 7206   4753                JMS I ACNEG4         /MAKE FAC POSITIVE
 72Ø7   1333                TAD CMINUS
 721Ø   1334                TAD CPLUS
 7211   3743                DCA I BUFADD         /SIGN (MINUS OR SPACE) INTO BUFFER
 7212   4736                JMS I AMNT3          /IS FAC = Ø?
 7213   524Ø                JMP FØ               /YES
 7214   1Ø44    RANGE,      TAD EXP              /NO. MAKE FAC INTO FAC'*(1Ø**M),
 7215   745Ø                SNA                  /WHERE Ø.1 < FAC' < 1.Ø
 7216   5235                JMP ROUND
 7217   77ØØ                SMA CLA
 722Ø   523Ø                JMP REDUCE
 7221   44Ø7                JMS I ØØØ7           /MULTIPLY FAC BY 1Ø
 7222   3637                FMPY I TEN1
 7223   ØØØØ                FEXT
 7224   724Ø                CLA CMA
 7225   137Ø                TAD READIN
 7226   337Ø                DCA READIN           /DECREMENT COUNTER
 7227   5214                JMP RANGE
 723Ø   4651    REDUCE,     JMS I DVTEN1         /DIVIDE FAC BY 1Ø
 7231   237Ø                ISZ READIN           /INCREMENT COUNTER
 7232   1Ø44                TAD EXP
 7233   774Ø                SMA SZA CLA          /IS FAC' IN RANGE?
 7234   523Ø                JMP REDUCE           /NO, CONTINUE
 7235   4754    ROUND,      JMS I RNDOT          /YES. ROUNDOFF OUTPUT
 7236   5243                JMP .+5
 7237   75ØØ    TEN1,       TEN
 724Ø   7Ø4Ø    FØ,         CMA                  /SET VALUES FOR FAC = Ø
 7241   3Ø44                DCA EXP
 7242   237Ø                ISZ READIN
 7243   1343                TAD BUFADD
 7244   3Ø15                DCA AUT1
 7245   4755                JMS I DECML
 7246   1343                TAD BUFADD
 7247   3Ø15                DCA AUT1
 725Ø   137Ø                TAD READIN
 7251   751Ø    DVTEN1,     SPA
 7252   72ØØ                CLA
 7253   1Ø54                TAD NDEC
 7254   7Ø41                CMA IAC
 7255   1Ø55                TAD NDIG
 7256   751Ø                SPA
 7257   5321                JMP FRMERR           /DATA WILL NOT FIT FORMAT
 726Ø   745Ø                SNA
 7261   527Ø                JMP .+7
 7262   7Ø41                CMA IAC              /OUTPUT LEADING SPACES
 7263   3Ø53                DCA LOC3
 7264   1335                TAD CSPCE
 7265   4337                JMS OUTT
```

```
7266  2053            ISZ LOC3
7267  5264            JMP .-3
7270  1743            TAD I BUFADD     /OUTPUT SIGN
7271  4463            JMS I TYPE
7272  1370            TAD READIN
7273  7510            SPA
7274  5312            JMP NEGEXP
7275  7450            SNA              /FOR FAC > 0.1: OUTPUT DIGITS
7276  5305            JMP .+7
7277  7041            CMA IAC
7300  3044            DCA EXP
7301  4360            JMS GETT
7302  4337            JMS OUTT
7303  2044            ISZ EXP
7304  5301            JMP .-3
7305  1356            TAD CPER         /DECIMAL POINT
7306  4463            JMS I TYPE
7307  4360            JMS GETT
7310  4337            JMS OUTT
7311  5307            JMP .-2
7312  3044  NEGEXP,   DCA EXP          /FOR FAC < 0.1: OUTPUT DIGITS
7313  1356            TAD CPER
7314  4463            JMS I TYPE
7315  4337            JMS OUTT         /OUTPUT ZEROES RIGHT OF DEC. PT.
7316  2044            ISZ EXP
7317  5315            JMP .-2
7320  5307            JMP NEGEXP-3
7321  1054  FRMERR,   TAD NDEC
7322  7700            SMA CLA          /CAN LEFT-SHIFTED FORMAT FIT?
7323  5327            JMP .+4          /YES
7324  7144            CLL CMA RAL      /NO: OUTPUT DECIMAL POINTS
7325  4337            JMS OUTT
7326  5324            JMP .-2
7327  7240            CLA CMA          /SHIFT FORMAT ONE SPACE LEFT
7330  1054            TAD NDEC
7331  3054            DCA NDEC
7332  5250            JMP DVTEN1-1
7333  0015  CMINUS,   0255-0240
7334  0240  CPLUS,    0240
7335  7760  CSPCE,    0240-0260
7336  6655  AMNT3,    AMOUNT
7337  0000  OUTT,     0                /OUTPUT DIGIT IN AC
7340  1357            TAD CZERO
7341  4463            JMS I TYPE
7342  2051            ISZ LOC1         /IS FORMAT FILLED?
7343  5737  BUFADD,   JMP I OUTT
7344  2053            ISZ LOC3         /YES: WAS FAC = 0?
7345  5350            JMP .+3          /NO
7346  1357            TAD CZERO        /YES: OUTPUT A ZERO
7347  4463            JMS I TYPE
7350  1061            TAD SWIT1        /IS CR-LF WANTED?
7351  7640            SZA CLA
7352  4464            JMS I CRLF       /YES
7353  5600  ACNEG4,   JMP I OUTPVT
7354  5512  RNDOT,    RNDOUT
7355  7533  DECML,    DECMAL
```

```
7356  Ø256  CPER,     Ø256
7357  Ø26Ø  CZERO,    Ø26Ø
736Ø  ØØØØ  GETT,     Ø                 /GET DIGIT FROM BUFFER
7361  2Ø52            ISZ LOC2
7362  5366            JMP .+4           /GETS FIRST 6 DIGITS FROM BUFFER
7363  724Ø            CLA CMA           /GIVES Ø FOR ALL FURTHER DIGITS
7364  3Ø52            DCA LOC2          /REQUESTED
7365  576Ø            JMP I GETT
7366  1415            TAD I AUT1
7367  576Ø            JMP I GETT
737Ø  ØØØØ  READIN,   Ø
7371  6Ø31            KSF
7372  5371            JMP .-1
7373  6Ø36            KRB
7374  6Ø46            TLS
7375  6Ø41            TSF
7376  5375            JMP .-1
7377  577Ø            JMP I READIN

            *74ØØ
74ØØ  ØØØØ  INPUT,    Ø                 /INPUT TO FAC
74Ø1  724Ø            CLA CMA
74Ø2  3Ø52            DCA LOC2          /PERIOD SWITCH SET TO 7777
74Ø3  1274            TAD C35
74Ø4  3Ø44            DCA EXP
74Ø5  3Ø45            DCA HORD
74Ø6  3Ø46            DCA LORD
74Ø7  3Ø47            DCA OVER2
741Ø  7Ø4Ø            CMA
7411  3Ø53            DCA LOC3          /SIGN SWITCH SET TO 7777
7412  3Ø15            DCA AUT1
7413  4315            JMS IN            /GET CHARACTER
7414  1Ø51            TAD LOC1
7415  1275            TAD MSPCE
7416  745Ø            SNA
7417  5213            JMP .-4           /IGNORE LEADING SPACES
742Ø  1276            TAD MPLUS
7421  745Ø            SNA
7422  5227            JMP DATA          /PLUS SIGN; DIGIT NEXT
7423  1277            TAD MMINUS
7424  764Ø            SZA CLA           /MINUS SIGN?
7425  523Ø            JMP DATA+1        /NO
7426  3Ø53            DCA LOC3          /YES; SET SIGN SWITCH TO Ø
7427  4315  DATA,     JMS IN            /GET CHARACTER
743Ø  1Ø51            TAD LOC1
7431  13Ø3            TAD M272
7432  75ØØ            SMA
7433  5247            JMP NODIG         /NOT A DIGIT
7434  13Ø4            TAD C1Ø
7435  751Ø            SPA
7436  5247            JMP NODIG         /NOT A DIGIT
7437  3665            DCA I ACNEG5      /DIGIT
744Ø  2Ø15            ISZ AUT1
7441  4775            JMS I MUL1Ø       /CONTINUE CONVERSION TO BINARY
7442  1Ø45            TAD HORD
7443  Ø247            AND NODIG
```

```
7444  7640              SZA CLA
7445  7402              HLT                     /TOO MANY DIGITS INPUT: NUMBER
7446  5227              JMP DATA                /MUST BE LESS THAN 2,147,483,648
7447  7600  NODIG,      7600
7450  1051              TAD LOC1
7451  1305              TAD MPER
7452  7640              SZA CLA                 /DECIMAL POINT?
7453  5257              JMP FIN                 /NO: TERMINATE
7454  3015              DCA AUT1                /YES: RESET COUNTER
7455  3052              DCA LOC2                /RESET PERIOD SWITCH TO 0
7456  5227              JMP DATA                /DEC. PT. IS NOT A TERMINATOR
7457  4732  FIN,        JMS I DNORM3
7460  2053              ISZ LOC3                /GIVE FAC PROPER SIGN
7461  4665              JMS I ACNEG5
7462  2052              ISZ LOC2                /HAS DEC. PT. BEEN INPUT?
7463  1015              TAD AUT1                /YES
7464  7450              SNA                     /NO: IT FOLLOWS LAST DIGIT
7465  5600  ACNEG5,     JMP I INPVT             /DONE
7466  7041              CIA                     /DIGITS TO RIGHT OF DEC. PT.
7467  3315              DCA IN                  /DIVIDE BY 10 APPROPRIATE
7470  4310              JMS DIVTEN              /NUMBER OF TIMES
7471  2315              ISZ IN
7472  5270              JMP .-2
7473  5600              JMP I INPVT
7474  0043  C35,        0043
7475  7540  MSPCE,      -240
7476  7765  MPLUS,      240-253
7477  7776  MMINUS,     253-255
7500  0004  TEN,        0004
7501  2400              2400
7502  0000              0000
7503  7506  M272,       -272
7504  0012  C10,        272-260
7505  7522  MPER,       -256
7506  7401  MRBOUT,     -377
7507  0162  MCR,        377-215
7510  0000  DIVTEN,     0                       /DIVIDE FAC BY 10
7511  4407              JMS I 0007              /DO NOT RELOCATE
7512  4300              FDIV TEN
7513  0000              FEXT
7514  5710              JMP I DIVTEN
7515  0000  IN,         0                       /READ A CHARACTER
7516  4462              JMS I READ
7517  3051              DCA LOC1
7520  1051              TAD LOC1
7521  1306              TAD MRBOUT
7522  7450              SNA                     /RUBOUT?
7523  5201              JMP INPVT+1             /YES: RESTART INPUT
7524  1307              TAD MCR
7525  7640              SZA CLA                 /CARRIAGE RETURN?
7526  5715              JMP I IN                /NO
7527  1771              TAD I C212P             /YES: OUTPUT LINE-FEED
7530  4463              JMS I TYPE              /THIS ALSO WILL TERMINATE INPUT
7531  5715              JMP I IN
7532  6400  DNORM3,     DNORM
```

```
7533  0000  DECMAL,   0
7534  1044            TAD EXP          /SHIFT SO EXP = 0 AND DEC. PT. IS
7535  7440            SZA              /AT LEFT OF BIT 0
7536  5341            JMP .+3
7537  4772            JMS I LSHF
7540  5346            JMP .+6
7541  7001            IAC
7542  3773            DCA I AMNT4
7543  7120            CLL CML
7544  2044            ISZ EXP
7545  4774            JMS I RSHFT2
7546  3665            DCA I ACNEG5
7547  1376            TAD N6           /SET COUNTER
7550  3044            DCA EXP
7551  4775            JMS I MUL10      /CALCULATE FIRST DIGIT
7552  7440            SZA              /IS IT 0?
7553  5360            JMP .+5          /NO
7554  7240            CLA CMA          /YES: IGNORE THIS AND ADJUST
7555  1462            TAD I READ       /TO CALCULATE 6 MORE DIGITS
7556  3462            DCA I READ
7557  4775            JMS I MUL10      /CALCULATE NEXT DIGIT
7560  3415            DCA I AUT1       /STORE DIGIT IN BUFFER
7561  2044            ISZ EXP
7562  5357            JMP .-3
7563  1055            TAD NDIG         /DIGITS CALCULATED: SET COUNTERS
7564  7041            CMA IAC
7565  3051            DCA LOC1
7566  1377            TAD N7
7567  3052            DCA LOC2
7570  3053            DCA LOC3
7571  5733  C2120,    JMP I DECMAL
7572  6455  LSHF,     LSHFT
7573  6655  AMNT4,    AMOUNT
7574  6670  RSHFT2,   RSHFT
7575  6555  MUL10,    X10
7576  7772  N6,       -6
7577  7771  N7,       -7
```

```
ACNEG     5600      DNORM     6400      MSPCE     7475      SERSW     5664
ACNEG1    6033      DNORM1    6260      MUL       7151      SETEV     5641
ACNEG2    6255      DNORM2    7150      MULSW     6265      SETEVN    6042
ACNEG3    6452      DNORM3    7532      MULT      6257      SETOD     6065
ACNEG4    7353      DOS       7171      MUL10     7575      SETODD    5652
ACNEG5    7465      DVTEN1    7251      M1        7055      SHIFT     6710
ACIH      0041      DVTEN2    5515      M272      7503      SIGN      6261
ACIL      0042      DVX       6323      M6        5550      SNREG     6300
ADD       7144      EXIT0     6445      NBRHI     0057      SNSW      6272
ADDOP     6534      EXIT1     6370      NBRLO     0060      SQARE     6772
ADDUP     6732      EXP       0044      NBRX      0056      SQROT     7156
ALIGN     6641      EX1       0040      NDEC      0054      SUB       7143
AMNT1     6375      FACNEG    6372      NDIG      0055      SWIT1     0061
AMNT2     6453      FCOS      6034      NEG       7060      TABLE1    6733
AMNT3     7336      FEXP      5620      NEGEXP    7312      TABLE2    5761
AMNT4     7573      FIN       7457      NEGOP     6301      TABL1     7056
AMNT5     5576      FIXX      5553      NODIG     7447      TABL2     7057
AMOUNT    6655      FLAD      7147      NRM       7145      TAG1      6730
ARCTN     6066      FLOG      5675      N6        7576      TAG2      6731
AUT1      0015      FLOWT     5753      N7        7577      TEN       7500
BIGG      6116      FPNT      7000      OFFSET    6645      TEN1      7237
BUFADD    7343      FRMERR    7321      OK        6416      TERM      6052
CMINUS    7333      FSIN      6000      ONE       5666      TRMSW     6057
CN12      6771      F0        7240      OUTPVT    7200      TR1       5737
CN23      6377      GET       7100      OUTT      7337      TR2       5742
CPER      7356      GETOP     6630      OUT0      5546      TR3       5745
CPLUS     7334      GETT      7360      OVER1     0043      TR4       5750
CRLF      0064      HALFPI    6172      OVER2     0047      TS1       6144
CRLFRT    5724      HORD      0045      PUT       7107      TS2       6177
CSPCE     7335      IN        7515      QUOL      0050      TS3       6171
CZERO     7357      INPVT     7400      RANGE     7214      TS4       6176
C10       7504      INVRT     6137      READ      0062      TWO       6134
C212      5733      INVRT1    5665      READIN    7370      TYPE      0063
C212P     7571      LOC1      0051      READY     6126      TYPOUT    7172
C215      5732      LOC2      0052      REDUCE    7230      UNE       5551
C24       6727      LOC3      0053      RNDOFF    6473      UNFLW     6511
C35       7474      LOGG2     5670      RNDOT     7354      UNFLW1    6374
DADD      6600      LORD      0046      RNDOUT    5512      UNFLW2    7154
DATA      7427      LSHF      7572      ROUND     7235      UNO       6133
DECMAL    7533      LSHFT     6455      RSHFT     6670      X         5734
DECML     7355      MCR       7507      RSHFT1    6454      XMP1      5642
DEUX      5673      MMINUS    7477      RSHFT2    7574      XX        6175
DIV       6302      MNSBR     7142      RSHFT3    5577      XXMP1     6051
DIVSW     6376      MPER      7505      SAV1      7141      XXX       5552
DIVTEN    7510      MPLUS     7476      SBR       7120      X10       6555
DMPY      6743      MPLY      7155      SERIES    6145
DMULT     6200      MRBOUT    7506      SERIEZ    5674
```

DISTRIBUTION LIST

No. of Copies To

1 Office of the Under Secretary of Defense for
Research and Engineering, The Pentagon,
Washington, DC 20301

12 Commander, Defense Technical Information Center,
Cameron Station, Building 5, 5010 Duke Street,
Alexandria, VA 22314

1 Battelle Columbus Laboratories, Metals and
Ceramics Information Center, 505 King Avenue,
Columbus, OH 43201

Deputy Chief of Staff for Research, Development,
and Acquisition, Headquarters, Department of the
Army, Washington, DC 20301
1 ATTN: DAMA-ARZ

Commander, Army Research Office, P.O. Box 12211,
Research Triangle Park, NC 27709
1 ATTN: Information Processing Office

Commander, U.S. Army Materiel Development and
Readiness Command, 5001 Eisenhower Avenue,
Alexandria, VA 22333
1 ATTN: DRCLDC

Commander, U.S. Army Materiel Systems Analysis
Activity, Aberdeen Proving Ground, MD 21005
1 ATTN: DRXSY-MP, H. Cohen

Commander, U.S. Army Electronics Research and
Development Command, Fort Monmouth, NJ 07703
1 ATTN: DELSD-L
1 DELSD-E

Commander, U.S. Army Missile Command, Redstone
Arsenal, AL 35809
1 ATTN: DRSMI-RKP, J. Wright, Bldg. 7574
4 DRSMI-TB, Redstone Scientific Information
Center
1 DRSMI-RLM
1 Technical Library

Commander, U.S. Army Armament Research and
Development Command, Dover, NJ 07801
2 ATTN: Technical Library
1 DRDAR-SCM, J. D. Corrie
1 DRDAR-QAC-E
1 DRDAR-LCA, Mr. Harry E. Pebly, Jr.,
PLASTEC, Director

Commander, U.S. Army Natick Research and
Development Laboratories, Natick, MA 01760
1 ATTN: Technical Library

Commander, U.S. Army Satellite Communications
Agency, Fort Monmouth, NJ 07703
1 ATTN: Technical Document Center

Commander, U.S. Army Tank-Automotive Command,
Warren, MI 48090
1 ATTN: DRSTA-RKA
2 DRSTA-UL, Technical Library

Commander, White Sands Missile Range, NM 88002
1 ATTN: STEWS-WS-VT

No. of Copies To

President, Airborne, Electronics and Special
Warfare Board, Fort Bragg, NC 28307
1 ATTN: Library

Director, U.S. Army Ballistic Research
Laboratory, Aberdeen Proving Ground, MD 21005
1 ATTN: DRDAR-TSB-S (STINFO)

Commander, Dugway Proving Ground, Dugway, UT
84022
1 ATTN: Technical Library, Technical Information
Division

Commander, Harry Diamond Laboratories, 2800
Powder Mill Road, Adelphi, MD 20783
1 ATTN: Technical Information Office

Chief, Benet Weapons Laboratory, LCWSL, USA
ARRADCOM, Watervliet, NY 12189
1 ATTN: DRDAR-LCB-TL
1 Dr. T. Davidson
1 Mr. D. P. Kendall

Commander, U.S. Army Foreign Science and
Technology Center, 220 7th Street, N. E.,
Charlottesville, VA 22901
1 ATTN: Military Tech, Mr. Marley

Commander, U.S. Army Aeromedical Research Unit,
P.O. Box 577, Fort Rucker, AL 36360
1 ATTN: Technical Library

Director, Eustis Directorate, U.S. Army Air
Mobility Research and Development Laboratory,
Fort Eustis, VA 23604
1 ATTN: Mr. J. Robinson, DAVDL-E-MOS (AVRADCOM)

U.S. Army Aviation Training Library, Fort
Rucker, AL 36360
1 ATTN: Building 5906-5907

Commander, U.S. Army Agency for Aviation Safety,
Fort Rucker, AL 36362
1 ATTN: Technical Library

Commander, USACDC Air Defense Agency, Fort
Bliss, TX 79916
1 ATTN: Technical Library

Commander, U.S. Army Engineer School, Fort
Belvoir, VA 22060
1 ATTN: Library

Commander, U.S. Army Engineer Waterways
Experiment Station, Vicksburg, MS 39180
1 ATTN: Research Center Library

Commander, U.S. Army Environmental Hygiene
Agency, Edgewood Arsenal, MD 21010
1 ATTN: Chief, Library Branch

Technical Director, Human Engineering
Laboratories, Aberdeen Proving Ground, MD 21005
1 ATTN: Technical Reports Office

No. of Copies	To
	Commandant, U.S. Army Quartermaster School, Fort Lee, VA 23801
1	ATTN: Quartermaster School Library
	Commander, U.S. Army Radio Propagation Agency, Fort Bragg, NC 28307
1	ATTN: SCCR-2
	Naval Research Laboratory, Washington, DC 20375
1	ATTN: Dr. J. M. Krafft - Code 5830
2	Dr. G. R. Yoder - Code 6384
	Chief of Naval Research, Arlington, VA 22217
1	ATTN: Code 471
	Commander, U.S. Air Force Wright Aeronautical Laboratories, Wright-Patterson Air Force Base, OH 45433
2	ATTN: AFWAL/MLSE, E. Morrissey
1	AFWAL/MLC
1	AFWAL/MLLP, M. Forney Jr.
1	AFWAL/MLBC, Mr. Stanley Schulman
	National Aeronautics and Space Administration, Washington, DC 20546
1	ATTN: Mr. B. G. Achhammer
1	Mr. G. C. Deutsch - Code RW
	National Aeronautics and Space Administration, Marshall Space Flight Center, Huntsville, AL 35812
1	ATTN: R. J. Schwinghammer, EH01, Dir, M&P Lab
1	Mr. W. A. Wilson, EH41, Bldg. 4612
1	Ship Research Committee, Maritime Transportation Research Board, National Research Council, 2101 Constitution Ave., N. W., Washington, DC 20418
1	Librarian, Materials Sciences Corporation, Blue Bell Campus, Merion Towle House, Blue Bell, PA 19422
	Panametrics, 221 Crescent Street, Waltham, MA 02154
1	ATTN: Mr. K. A. Fowler
1	The Charles Stark Draper Laboratory, 68 Albany Street, Cambridge, MA 02139
	Wyman-Gordon Company, Worcester, MA 01601
1	ATTN: Technical Library
	Lockheed-Georgia Company, 86 South Cobb Drive, Marietta, GA 30063
1	ATTN: Materials and Processes Engineering Dept. 71-11, Zone 54
	General Dynamics, Convair Aerospace Division, P.O. Box 748, Fort Worth, TX 76101
1	ATTN: Mfg. Engineering Technical Library
1	Mechanical Properties Data Center, Belfour Stulen Inc., 13917 W. Bay Shore Drive, Traverse City, MI 49684
1	Dr. Robert S. Shane, Shane Associates, Inc., 7821 Carrleigh Parkway, Springfield, VA 22152
1	Mr. R. J. Zentner, EAI Corporation, 190 Thomas Johnson Drive, Suite 16, Frederick, MD 21701
	Director, Army Materials and Mechanics Research Center, Watertown, MA 02172
2	ATTN: DRXMR-PL Author

Army Materials and Mechanics Research Center
Watertown, Massachusetts 02172
FPP1, A FLOATING POINT PACKAGE FOR
PDP-8 COMPUTERS
Christopher B. Walker

Technical Report AMMRC TR 83-47, August 1983, 32 pp-illus-tables, D/A Project 1L161102AH42,

AD____________
UNCLASSIFIED
UNLIMITED DISTRIBUTION

Key Words

Computer program documentation
Computer programs
Computers

This report describes a 23-bit floating point package for PDP-8 computers developed from the DEC YQ4B package that retains most of the capabilities of YQ4B while requiring only 1101 words storage.

Army Materials and Mechanics Research Center
Watertown, Massachusetts 02172
FPP1, A FLOATING POINT PACKAGE FOR
PDP-8 COMPUTERS
Christopher B. Walker

Technical Report AMMRC TR 83-47, August 1983, 32 pp-illus-tables, D/A Project 1L161102AH42,

AD____________
UNCLASSIFIED
UNLIMITED DISTRIBUTION

Key Words

Computer program documentation
Computer programs
Computers

This report describes a 23-bit floating point package for PDP-8 computers developed from the DEC YQ4B package that retains most of the capabilities of YQ4B while requiring only 1101 words storage.

Army Materials and Mechanics Research Center
Watertown, Massachusetts 02172
FPP1, A FLOATING POINT PACKAGE FOR
PDP-8 COMPUTERS
Christopher B. Walker

Technical Report AMMRC TR 83-47, August 1983, 32 pp-illus-tables, D/A Project 1L161102AH42,

AD____________
UNCLASSIFIED
UNLIMITED DISTRIBUTION

Key Words

Computer program documentation
Computer programs
Computers

This report describes a 23-bit floating point package for PDP-8 computers developed from the DEC YQ4B package that retains most of the capabilities of YQ4B while requiring only 1101 words storage.

Army Materials and Mechanics Research Center
Watertown, Massachusetts 02172
FPP1, A FLOATING POINT PACKAGE FOR
PDP-8 COMPUTERS
Christopher B. Walker

Technical Report AMMRC TR 83-47, August 1983, 32 pp-illus-tables, D/A Project 1L161102AH42,

AD____________
UNCLASSIFIED
UNLIMITED DISTRIBUTION

Key Words

Computer program documentation
Computer programs
Computers

This report describes a 23-bit floating point package for PDP-8 computers developed from the DEC YQ4B package that retains most of the capabilities of YQ4B while requiring only 1101 words storage.